

IN THE CLAIMS

Please amend the above-identified application as follows:

1. (Currently Amended) A program execution apparatus that determines an execution sequence of tasks and executes the tasks according to the execution sequence, each task being given a target completion time before which execution of the task is to be completed, the program execution apparatus comprising:

a storing unit operable to store at least one identifier of at least one task that is already in existence, at a memory position therein determined based upon a plurality of types of priorities set for the task, the plurality of types of priorities having a hierarchical relationship;

a receiving unit operable to receive an identifier of a new task and a plurality of types of priorities set for the new task;

a writing unit operable to write the identifier received by the receiving unit, at a memory position in the storing unit determined based upon the plurality of types of priorities received by the receiving unit; and

a determining unit operable to determine an execution sequence of the tasks whose identifiers are stored in the storing unit, according to an arrangement sequence of the identifiers in the storing unit, wherein

at least a first-type priority and a second-type priority are set for each task as the plurality of types of priorities, the second-type priority being lower in a hierarchy than the first-type priority,

the storing unit stores a plurality of groups each including identifiers of tasks for which a same value is set as a first-type priority, the plurality of groups being arranged at memory positions in the storing unit in an arrangement sequence determined in an order of the value of a

first-type priority set for each group, the identifiers of the tasks in each group being arranged at memory positions in each group in the storing unit in an arrangement sequence determined based upon a second-type priority set for each task in the group,

the receiving unit receives the identifier of the new task, and a value of a first-type priority and a value of a second-type priority that are set for the new task,

the writing unit writes the received identifier, at the memory position in the storing unit determined based upon the received value of the second-type priority in a group including identifiers of tasks for which a first-type priority that is the same as the received value of the first-type priority is set, and

the determining unit determines the execution sequence of the tasks whose identifiers are stored in the storing unit, according to the arrangement sequence of the identifiers in the storing unit determined based upon a first-type priority and a second-type priority set for each task.

2. (Currently Amended) The program execution apparatus of Claim 1, wherein

~~a first type priority and a second type priority are set for each task as the plurality of types of priorities, the first-type priority being is a target completion time before which execution of the task for which the first-type priority is set is to be completed, the second-type priority being lower in a hierarchy than the first-type priority,~~

the storing unit stores a plurality of groups each including identifiers of tasks for which a same target completion time is set as a first-type priority, the plurality of groups being arranged at memory positions in the storing unit in an arrangement sequence determined in an order of a target completion time set for each group, ~~the identifiers of the tasks in each group being~~

~~arranged at memory positions in each group in the storing unit in an arrangement sequence
determined based upon a second type priority set for each task in the group,~~

the receiving unit receives ~~the identifier of the new task,~~ and a target completion time as
[[a]] the first-type priority and a second type priority that are set for the new task, and

the writing unit writes the received identifier, at the memory position in the storing unit
determined based upon the received second-type priority in a group including identifiers of tasks
for which a first-type priority that ~~[[is]]~~ has the same target completion time as the received first-
type priority is set, ~~and~~

~~the determining unit determines the execution sequence of the tasks whose identifiers are
stored in the storing unit, according to the arrangement sequence of the identifiers in the storing
unit determined based upon a first type priority and a second type priority set for each task.~~

3. (Original) The program execution apparatus of Claim 2, wherein

the storing unit stores, for a beginning group that includes identifiers of tasks for which a
target completion time closest to a present time is set as a first-type priority and that is arranged
at a beginning of the arrangement sequence of the plurality of groups, the closest target
completion time as the first-type priority, and stores, for each of groups that follow the beginning
group in the arrangement sequence, a time period between (a) a target completion time set for
tasks whose identifiers are included in the group and (b) a target completion time set for tasks
whose identifiers are included in a group that precedes the group in the arrangement sequence,
and

the writing unit compares the target completion time received by the receiving unit with
the closest target completion time, and (i) when finding a match, writes the received identifier, at
the memory position in the storing unit determined based upon the received second-type priority

in the beginning group, and (ii) when not finding a match, further compares, for each of the groups that follow the beginning group, (c) the target completion time received by the receiving unit with (d) a time at which the time period stored for the group elapses from a target completion time set for a group that precedes the group in the arrangement sequence, and when finding a match, writes the received identifier at the memory position in the storing unit determined based upon the received second-type priority in the group.

4. (Original) The program execution apparatus of Claim 2, wherein
a lowest value of a first-type priority is a maximum value that can be expressed by a predetermined bit sequence constituting a target completion time,
the storing unit stores a last group including identifiers of tasks for which a target completion time constituted by a predetermined bit sequence expressing the maximum value is set, at a memory position that is a last of the arrangement sequence of the plurality of groups, determined based upon the target completion time,
the receiving unit receives the identifier of the new task, and the target completion time constituted by the bit predetermined sequence expressing the maximum value and the second-type priority that are set for the new task, and
the writing unit writes the received identifier, at the memory position in the storing unit determined based upon the received second-type priority in the last group.

5. (Original) The program execution apparatus of Claim 2, wherein
the storing unit further stores, for each group, a total execution time period that is predicted to take to execute all tasks whose identifiers are included in the group,
the receiving unit further receives an execution time period that is predicted to take to execute the new task, and

the program execution apparatus further includes a judging unit operable to judge whether a time at which a time period obtained by adding the total execution time period and the execution time period received by the receiving unit elapses from a present time is before the target completion time set for the new task, and when judging negatively, output a reject signal indicating to reject execution of the new task,

wherein when the judging unit judges affirmatively, the writing unit writes the received identifier, at the memory position in the storing unit determined based upon the received second-type priority in a group including identifiers of tasks for which a first-type priority that is the same as the received first-type priority is set.

6. (Original) The program execution apparatus of Claim 5, wherein

the judging unit selects one of groups that follow a group in the arrangement sequence of the plurality of groups, and judges whether a time at which a total execution time period for the selected group elapses from a present time is before a target completion time set for tasks whose identifiers are included in the selected group, the judging unit repeating the selection and judgment processes on each of the following groups, and

when the judging unit judges affirmatively for all of the following groups, the writing unit writes the received identifier, at the memory position in the storing unit determined based upon the received second-type priority in a group including identifiers of tasks for which a first-type priority that is the same as the received first-type priority is set, and

when the judging unit judges negatively for any of the following groups, the writing unit outputs a reject signal indicating to reject execution of the new task.

7. (Original) The program execution apparatus of Claim 1, wherein

a first-type priority and a second-type priority are set for each task, the first-type priority being a value obtained by multiplying predetermined time-units of a target completion time set for the task by an integer, the second-type priority being a value of a remaining time-unit of the target completion time that is a smaller unit than the predetermined time-units,

the storing unit stores a plurality of groups each including identifiers of tasks for which a same first-type priority is set, the plurality of groups being arranged at memory positions in the storing unit in an arrangement sequence determined based upon a first-type priority set for each group, the identifiers of the tasks in each group being arranged at memory positions in the storing unit in an arrangement sequence determined based upon a second-type priority set for each task in the group,

the receiving unit receives the identifier of the new task, and a first-type priority and a second-type priority that are set for the new task,

the writing unit writes the received identifier, at a memory position in the storing unit determined based upon the received second-type priority in a group including identifiers of tasks for which a first-type priority that is the same as the received first-type priority is set, and

the determining unit determines the execution sequence of the tasks whose identifiers are stored in the storing unit, according to the arrangement sequence of the identifiers in the storing unit determined based upon a first-type priority and a second-type priority set for each task.

8. (Currently Amended) A task management method for use in a program execution apparatus that determines an execution sequence of tasks and executes the tasks according to the execution sequence, each task being given a target completion time before which execution of the task is to be completed, the program execution apparatus including a storing unit operable to store at least one identifier of at least one task that is already in existence, at a memory position

therein determined based upon a plurality of types of priorities set for the task, the plurality of types of priorities having a hierarchical relationship,

the task management method comprising:

a receiving step of receiving an identifier of a new task and a plurality of types of priorities set for the new task;

a writing step of writing the identifier received in the receiving step, at a memory position in the storing unit determined based upon the plurality of types of priorities received in the receiving step; and

a determining step of determining an execution sequence of the tasks whose identifiers are stored in the storing unit, according to an arrangement sequence of the identifiers in the storing unit, wherein

at least a first-type priority and a second-type priority are set for each task as the plurality of types of priorities, the second-type priority being lower in a hierarchy than the first-type priority, and

the storing unit stores a plurality of groups each including identifiers of tasks for which a same value is set as a first-type priority, the plurality of groups being arranged at memory positions in the storing unit in an arrangement sequence determined in an order of the value of a first-type priority set for each group, the identifiers of the tasks in each group being arranged at memory positions in each group in the storing unit in an arrangement sequence determined based upon a second-type priority set for each task in the group, and wherein

the receiving step receives the identifier of the new task, and a value of the first-type priority and a value of the second-type priority that are set for the new task,

the writing step writes the received identifier, at the memory position in the storing unit determined based upon the received value of the second-type priority in a group including identifiers of tasks for which the first-type priority that is the same as the received value of the first-type priority is set, and

the determining step determines the execution sequence of the tasks whose identifiers are stored in the storing unit, according to the arrangement sequence of the identifiers in the storing unit determined based upon the first-type priority and the second-type priority set for each task.

9. (Currently Amended) A computer-readable recording medium have stored thereon a task management program for use in a program execution apparatus, the computer-readable recording medium causing the program execution apparatus to determine ~~that determines~~ an execution sequence of tasks and ~~executes~~ execute the tasks according to the execution sequence, each task being given a target completion time before which execution of the task is to be completed, the program execution apparatus including a storing unit operable to store at least one identifier of at least one task that is already in existence, at a memory position therein determined based upon a plurality of types of priorities set for the task, the plurality of types of priorities having a hierarchical relationship,

the task management program comprising:

a receiving step of receiving an identifier of a new task and a plurality of types of priorities set for the new task;

a writing step of writing the identifier received in the receiving step, at a memory position in the storing unit determined based upon the plurality of types of priorities received in the receiving step; and

a determining step of determining an execution sequence of the tasks whose identifiers are stored in the storing unit, according to an arrangement sequence of the identifiers in the storing unit, wherein

at least a first-type priority and a second-type priority are set for each task as the plurality of types of priorities, the second-type priority being lower in a hierarchy than the first-type priority, and

the storing unit stores a plurality of groups each including identifiers of tasks for which a same value is set as a first-type priority, the plurality of groups being arranged at memory positions in the storing unit in an arrangement sequence determined in an order of the value of a first-type priority set for each group, the identifiers of the tasks in each group being arranged at memory positions in each group in the storing unit in an arrangement sequence determined based upon a second-type priority set for each task in the group, and wherein

the receiving step receives the identifier of the new task, and a value of the first-type priority and a value of the second-type priority that are set for the new task,

the writing step writes the received identifier, at the memory position in the storing unit determined based upon the received value of the second-type priority in a group including identifiers of tasks for which the first-type priority that is the same as the received value of the first-type priority is set, and

the determining step determines the execution sequence of the tasks whose identifiers are stored in the storing unit, according to the arrangement sequence of the identifiers in the storing unit determined based upon the first-type priority and the second-type priority set for each task.

10. (Canceled)

11. (Currently Amended) A mobile telephone that determines an execution sequence of tasks and executes the tasks according to the execution sequence, each task being given a target completion time before which execution of the task is to be completed, the mobile telephone comprising:

a storing unit operable to store at least one identifier of at least one task that is already in existence, at a memory position therein determined based upon a plurality of types of priorities set for the task, the plurality of types of priorities having a hierarchical relationship;

a receiving unit operable to receive an identifier of a new task and a plurality of types of priorities set for the new task;

a writing unit operable to write the identifier received by the receiving unit, at a memory position in the storing unit determined based upon the plurality of types of priorities received by the receiving unit; and

a determining unit operable to determine an execution sequence of the tasks whose identifiers are stored in the storing unit, according to an arrangement sequence of the identifiers in the storing unit, wherein

at least a first-type priority and a second-type priority are set for each task as the plurality of types of priorities, the second-type priority being lower in a hierarchy than the first-type priority,

the storing unit stores a plurality of groups each including identifiers of tasks for which a same value is set as a first-type priority, the plurality of groups being arranged at memory positions in the storing unit in an arrangement sequence determined in an order of the value of a first-type priority set for each group, the identifiers of the tasks in each group being arranged at

memory positions in each group in the storing unit in an arrangement sequence determined based upon a second-type priority set for each task in the group,

the receiving unit receives the identifier of the new task, and a value of a first-type priority and a value of a second-type priority that are set for the new task,

the writing unit writes the received identifier, at the memory position in the storing unit determined based upon the received value of the second-type priority in a group including identifiers of tasks for which a first-type priority that is the same as the received value of the first-type priority is set, and

the determining unit determines the execution sequence of the tasks whose identifiers are stored in the storing unit, according to the arrangement sequence of the identifiers in the storing unit determined based upon a first-type priority and a second-type priority set for each task.